Esame di Calcolatori Elettronici T
16 Gennaio 2019 (Ing. Informatica)

Esercizio 1

Progettare un sistema, basato sul processore DLX, dotato **di 512 MB di EPROM** mappata agli indirizzi bassi e **1 GB di RAM** mappata agli indirizzi alti.
Il sistema deve inviare delle informazioni **a quattro unità esterne con le quali è possibile comunicare attraverso il protocollo di handshake**. L'invio dei dati deve avvenire **esclusivamente quando almeno due unità esterne sono pronte** a ricevere le informazioni. Quando tale condizione è verificata il processore deve inviare, **contemporaneamente a tutte le unità esterne in grado di riceverlo**, il **byte più aggiornato letto da una porta** che comunica con l'esterno mediante il protocollo di handshake. Il **dato letto da tale porta** dovrà essere **memorizzato a FFFF0000h.** Le operazioni appena descritte debbono essere svolte dal processore nel modo più rapido possibile. La lettura dalla porta in input è prioritaria rispetto alla scrittura verso le porte in output. Infine, all'avvio, **nessun dato deve essere inviato alle unità esterne fintantoché non è stato letto almeno un byte** dalla porta in input.

- **Per prima cosa, descrivere sinteticamente la soluzione** che s'intende realizzare **e** indicare **chiaramente quali sono i dispositivi utilizzati e segnali di *chip-select***
- Progettare il sistema, **minimizzando le risorse necessarie** ed **evidenziando eventuali criticità**
- Indicare le espressioni di decodifica e il range di indirizzi di tutte le periferiche, le memorie e i segnali
- Scrivere il codice dell'interrupt *handler*
- Si faccia l'ipotesi che i registri da R25 a R29 possano essere utilizzati senza la necessità di doverli ripristinare durante l'esecuzione degli interrupt handler

Esercizio 2

a) In cosa consiste un *branch target buffer*?
b) In quale stadio agisce?
c) Come può essere reso più efficace in caso di loop?

Esercizio 3

a) Che problema mira a risolvere la tecnica del *delayed load*?
b) Come agisce?

MAPPING

EPROM 512 MB: 0x0000 0000 → 0x1FFFFFFF (4×128)

RAM 512 MB L: 0xC000 0000 → 0xDFFEFFFF (4×128)

RAM 512 MB H: 0xE000 0000 → 0xFFFEFFFF (4×128)

$CS\_EPROM\_0 = \overline{BA31}\ BA30\ BE0$

" _1 = " BE1

" _2 = " BE2

" _3 = " BE3

$CS\_RAM\_H\_0 = BA31\ BA29\ BE0$

" _1 = " BE1

" _2 = " BE2

" _3 = " BE3

$CS\_RAM\_L\_0 = BA31\ \overline{BA29}\ BE0$

" _1 = " BE1

" _2 = " BE2

" _3 = " BE3

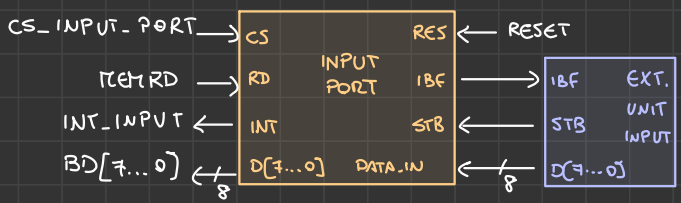
$CS\_INPUT = \overline{BA31}\ BA30\ BE0$

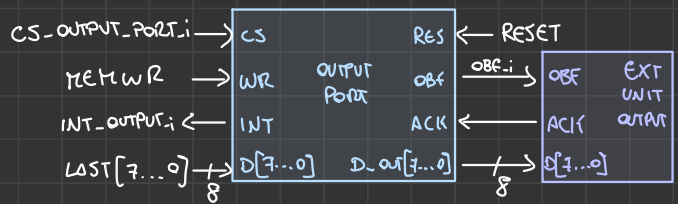
$CS\_OUTPUT = \overline{BA31}\ BA30\ BE1$

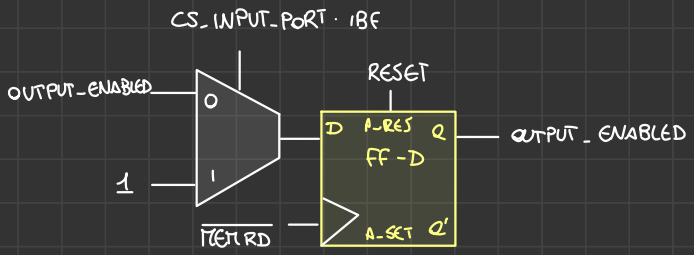
$CS\_READ\_INPUT\_STATUS = \overline{BA31}\ BA30\ BE2\ MEMRD$

$CS\_READ\_OUTPUT\_STATUS = \overline{BA31}\ BA30\ BE3\ MEMRD$

$ALTENO\_DUE = INT\_OUTPUT\_0 \cdot INT\_OUTPUT\_1 + INT\_OUTPUT\_0 \cdot INT\_OUTPUT\_2$
$INT\_OUTPUT\_0 \cdot INT\_OUTPUT\_3 + INT\_OUTPUT\_1 \cdot INT\_OUTPUT\_3$
$INT\_OUTPUT\_1 \cdot INT\_OUTPUT\_2 + INT\_OUTPUT\_2 \cdot INT\_OUTPUT\_3$

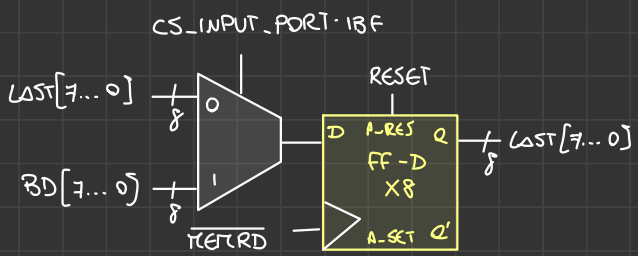


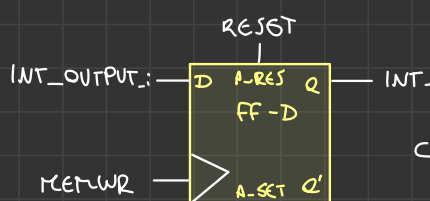




INT (TO DCX) = INT_INPUT + ALTENO_DUE · OUTPUT_ENABLED

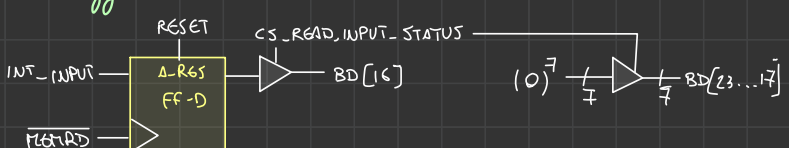
Rete per campionare l'ultimo dato letto da INPUT PORT

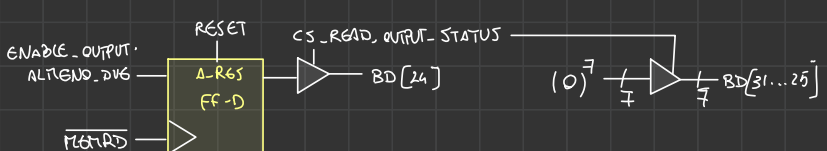




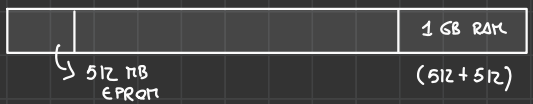
CS_OUTPUT_i = CS_OUTPUT_PORT · INT_OUTPUT_C_i

Campioniamo INT_INPUT e ALTENO_DUE · OUTPUT_ENABLED per leggerli nell'handler





CHIP SELECT



$CS\_EPROM\_0 = \overline{BA31\ BA30}\ BE0$

$\quad\quad\ \_1 = \quad\quad\ BE1$

$\quad\quad\ \_2 = \quad\quad\ BE2$

$\quad\quad\ \_3 = \quad\quad\ BE3$

$CS\_RAM\_L\_0 = BA31\ \overline{BA29}\ BE0$

$\quad\quad\ \_1 = \quad\quad\ BE1$

$\quad\quad\ \_2 = \quad\quad\ BE2$

$\quad\quad\ \_3 = \quad\quad\ BE3$

$CS\_RAM\_H\_0 = BA31\ BA29\ BE0$

$\quad\quad\ \_1 = \quad\quad\ BE1$

$\quad\quad\ \_2 = \quad\quad\ BE2$

$\quad\quad\ \_3 = \quad\quad\ BE3$

$CS\_INPUT\_PORT = \overline{BA31}\ BA30\ BE0$

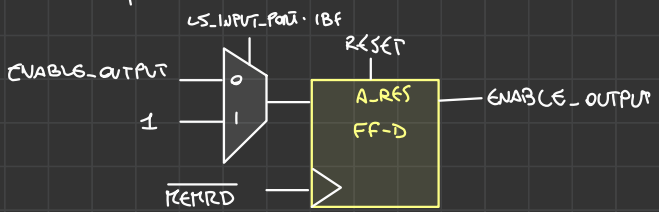
$CS\_OUTPUT\_PORT = \quad\quad\ BE1$

$CS\_READ\_INPUT\_STATUS = \quad\quad\ BE2\ \overline{MEMRD}$

$CS\_READ\_OUTPUT\_STATUS = \quad\quad\ BE3\ \overline{MEMRD}$

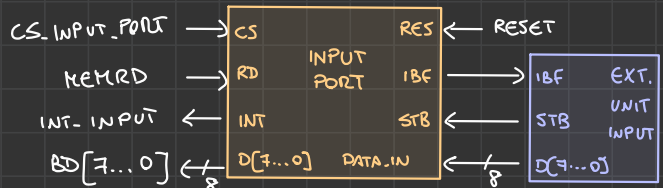
$$ALMENO\_DUE = INT\_OUT\_0 \cdot INT\_OUT\_1 + INT\_OUT\_0 \cdot INT\_OUT\_2 + INT\_OUT\_0 \cdot INT\_OUT\_3 + INT\_OUT\_1 \cdot INT\_OUT\_2 + INT\_OUT\_1 \cdot INT\_OUT\_3 + INT\_OUT\_2 \cdot INT\_OUT\_3$$

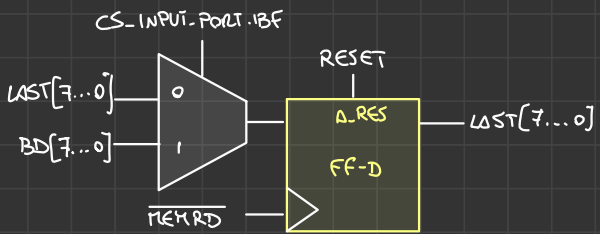
La rete seguente avrà uscita 0 fin quando non si avrà un trasferimento dalla porta in INPUT

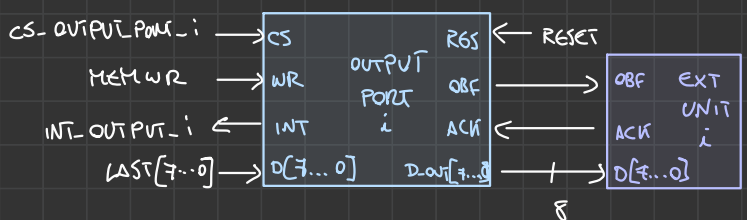


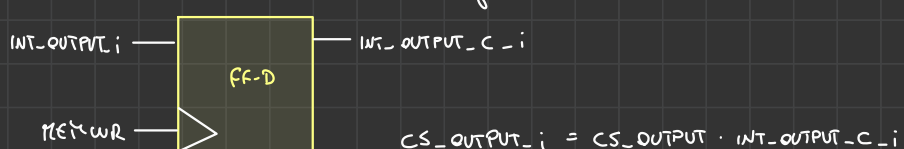
$$INT1\ TO\ DLX) = INT\_INPUT + ALMENO\_DUE \cdot ENABLE\_OUTPUT$$

PORTE DI I/O

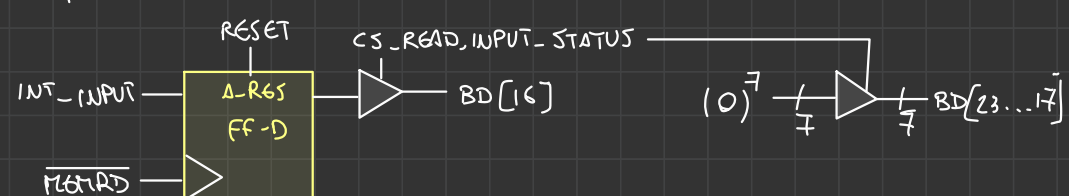


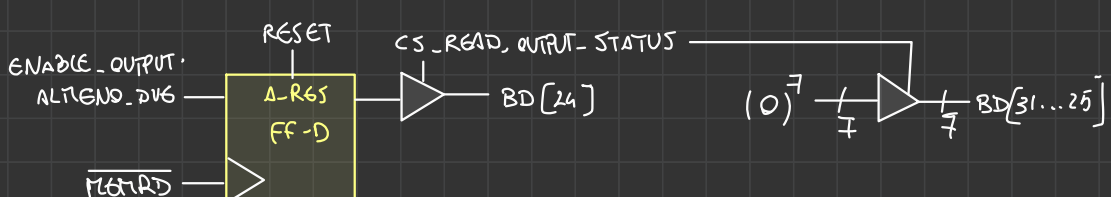






Compi qui anche INT_INPUT e ALTERNO_DVE · ENABLE_OUTPUT





CODICE

```
0h      LHI  R25, 0x4000
4h      LB   R26, 0x0002(R25)
8h      BEQZ R26, OUTPUT
Ch      LBU  R27, 0x0000(R25)
10h     LHI  R28, 0xFFFF
14h     SB   R27, 0x0000(R28)
18h OUTPUT: LBU  R26, 0x0003(R25)
1Ch         BEQZ R26, FINE
20h         SB   R26, 0x0001(R25)
24h FINE:   RFE
```